REMARKS

The Examiner rejected claims 1-36 as obvious (35 U.S.C. §103) over Harper (U.S. Patent No. 5,765,165). Applicants traverse for the following reasons.

Amended independent claims 1, 10, and 19 concern processing an input file in a file system, wherein the input file has an input file name, by: providing a data structure generated by applying a function to file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate file names used in the file system; applying a function to map the input file name to a value; and processing the data structure to determine whether there is a preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps, wherein two files that map to a same value according to the function are capable of having a same name.

Applicants amended claims 1, 10, and 19 to add the requirement of providing a data structure generated by applying a function to file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate file names used in the file system. The added requirement is disclosed in the Application on at least page 6, line 24 to page 7, line 14.

The Examiner cited col. 4, lines 21-29, FIG. 8, and col. 8, lines 20-35 as teaching the requirements of claims 1, 10, and 19. (Third Office Action, pgs. 2-4) The cited sections of Harper discuss how a hash function is applied to identifications of elements on a linked list to hash the identifiers of the elements to bits in a short byte block in memory . Indication is made that the element identifier is in the linked list used by setting the bit in the hash bit map corresponding to the hash of the element identifier to "1". When one wants to check whether the linked list has a duplicate of an element to add, the hash is applied to the identifier of the element to add to check whether the bit in the short byte block corresponding to the hashed identifier of the element to add is "1". If the entry in the hash bit map for hash of the identifier of the element to add is set to "1", then that identifier is a duplicate of one in the linked list, and a comparison of

the identifier to be added is made to each identifier in the linked list to find a duplicate. If a duplicate is found, that is reported.

Applicants submit that the cited sections of Harper nowhere teach or suggest the added claim requirement of providing a data structure generated by applying a function to file names in a file system to determine values corresponding to the file names, wherein the data structure indicates those values corresponding to the file names to indicate file names used in the file system. Instead, Harper discusses a byte in memory to identify the identifier of elements on a linked list, not file names in a file system as claimed.

The Examiner recognized that Harper did not explicitly teach using a hash to identify duplicate file names in a file system, but found that such a modification of Harper would be obvious to apply to file names in a file system. (Third Office Action, pg. 3) Applicants traverse.

The Manual of Patent Examination Procedure (MPEP) states that the "mere fact that references can be combined or modified does not render the resultant combination obvious unless the prior art also suggests the desirability of the combination." MPEP 2143.01, pg. 2100-124 (8th Ed., Aug. 2001). Here, the Examiner's proposed modification of the cited Harper is improper because the Examiner has not cited any part of Harper or other art that suggests a data structure applying a function, such as a hash function, to file names in a file system to indicate file names used in the file system. There simply is no teaching or suggestion anywhere in the cited art of using a hash for this claimed purpose of determining whether an input file name in a file system is already used in the file system.

Moreover, Harper teaches away from applying his hash to file names in a file system. Harper does its hash "by allocating a relatively short block of memory, such as one byte, and by hashing inode identifications to one of the bits of the block of memory." (Harper, col. 4, lines 1-5) The claimed data structure requires indication of hash values corresponding to file names in a file system and would require far more than one byte of memory allocated because a file system as known in the art must accommodate thousands of file names. See, Application, pg. 6, line 24 to pg. 7, line 14. Thus, Harper teaches away from applying the discussed hash technique for checking duplicates in a linked list to duplicates in a file system because Harper discusses that

his "present invention" allocates a short block of memory, such as one byte, as the hash bitmap used to indicate identifiers elements in a linked list.

The MPEP further states that:

> If the proposed modification or combination of the prior art would change the principle of operation of the prior art invention being modified, then the teachings of the references are not sufficient to render the claims prima facie obvious.

MPEP, Sec. 2143.02, pg. 125.

Applicants submit that in this case, the proposed modification of Harper to apply to file systems would change the principle of operation by requiring the allocation of memory for the hash bit map far exceeding the one byte Harper discusses in the "Summary of the Invention" section and far exceeding the hash needed to represent elements in a linked list. (Harper, col. 3, line 65 to col. 4, line 5)

For all the above reasons, the Examiner's proposed modification of Harper is improper because the cited art nowhere teaches or suggests the proposed modification and because such proposed modification teaches away from Harper, which discusses only the need of a smaller hash and hash bitmap to represent identifiers in a linked list.

Accordingly, for all the above reasons, amended claims 1, 10, and 19 are patentable over the cited art, which does not teach or suggest all the claim requirements.

Claims 2-9 and 28-30; 11-18 and 31-33; and 20-27 and 34-39 are patentable over the cited Harper because they depend from one of claims 1, 10, and 19, respectively, which are patentable over the cited combination for the reasons described above. Further claims discussed below provide additional grounds of patentability over the cited art.

Claims 4, 13, and 22 depend from claims 3, 12, and 13, which require that the data structure includes an entry for each possible integer value capable of being generated from the hash function. Claims 4, 13, and 22 further require that processing the data structure to determine whether there is a preexisting file comprises determining whether the entry for the integer value to which the input file name maps indicates the presence of one preexisting file mapping to the same integer value as the input file name.

The Examiner cited col. 8, lines 25-35 of Harper as teaching the additional requirements of these claims. (Third Office Action, pgs. 4-5) Applicants traverse.

As discussed, the cited col. 8 discusses determining whether the identifier of an element to add to a linked list is a duplicate of an element in the linked list. Nowhere does the cited Harper anywhere teach or suggest processing the data structure to determine whether there is a preexisting file in the file system having the same name as the input file name based on the entry for the integer value as claimed. Instead, Harper concerns duplicates for elements on a linked list and nowhere teaches, suggests or remotely using a function and data structure as claimed to determine whether an input file name has a same name as a preexisting file in the file system.

For these reasons, claims 4, 13, and 22 provide additional grounds of patentability over the cited combined art.

Claims 6, 15, and 24 depend from claims 1, 10, and 19 and further require applying the function to each file name in the file system to map each file name to one value and indicating in the data structure, for each file name, that there is one preexisting file for the value to which the file name maps.

The Examiner cited col. 8, lines 20-28 of Harper as teaching the additional requirements of these claims. (Third Office Action, pg. 5) Applicants traverse.

The cited Harper only discusses applying a hash to elements on a linked list. Nowhere does the cited Harper anywhere teach or suggest applying a function, or the hash of Harper, to each file name in a file system as claimed to indicate whether, for each file name, there is one preexisting file. Instead, as discussed, the cited Harper only discusses indicating in the hash bit map identifiers elements in a linked list.

For these reasons, the cited references do not teach or suggest, alone or in combination, the requirements of claims 6, 15, and 24.

Claims 7, 16, and 25 depend from claims 6, 15, and 24 and further require that the input file is the subject of an access request. Further, each file in the file system is scanned to determine if there is at least one preexisting file having the same name as the input file name if

there is one preexisting file in the file system having a name that maps, according to the function, to the same value to which the input file name maps.

The Examiner cited col. 8, lines 29-35 of Harper as teaching the additional requirements of these claims. (Third Office Action, pgs. 5-6) Applicants traverse.

The cited col. 8 discusses comparing the identifier to add to the identifiers of elements on a list once a potential duplicate is identified. Nowhere does the cited Harper anywhere teach, suggest or remotely mention that the input file is the subject of an access request.

Again, the Examiner is modifying prior art in a manner nowhere taught or suggested in the cited art. The proposed modifications are based solely on the claim requirements and not suggested in any cited art. Although different elements of the claims may be separately discussed in the cited art, such as a file system, hash functions, etc., nowhere does the cited Harper anywhere suggest the combination of these requirements as claimed to determine whether there is a preexisting file for a requested file in a file system.

Accordingly, the cited references do not teach or suggest, alone or in combination, the requirements of claims 7, 16, and 25.

Claims 8, 9, 17, 18, 26, and 27 depend from claims 7, 16 or 14 and provide further details on file system access operations based on using the claimed technique to determine whether there is a preexisting file having the same name as the input file subject to the access request. Because the Examiner has cited the same art in rejecting these claims (Third Office Action, pgs. 6-7), the cited combination fails to disclose the additional requirements of these claims in combination with the base and intervening claims from which they depend for the reasons discussed above, and because there is no suggestion in the cited art to modify Harper to use the discussed hash function with file names in a file system as claimed.

Claims 28, 31, and 34 depend from claims 1, 10, and 19 and further require searching the file system for one file having the same name as the input file name if the data structure indicates that one preexisting file has a name that maps, according to the function, to the same value to which the input file maps. An operation is performed if the file system includes one file having the same name as the input file.

The Examiner cited col. 8, lines 25-35 of Harper as teaching the additional requirements of these claims. (Third Office Action, pg. 7) Applicants traverse.

The cited col. 8 discusses comparing the identifier to add to the identifiers of elements on a list once a potential duplicate is identified. Nowhere does this cited col. 8 anywhere teach, suggest or remotely mention searching a file system for a preexisting file having the same name as the input file as claimed.

Accordingly, claims 28, 31, and 34 provide further grounds of distinction over the cited art.

Applicants further submit that claims 29, 30, 32, 33, 35, and 36 are also patentable over the cited art because they depend from claims 28, 31, and 34, which are patentable over the cited art for the reasons discussed above, and because the additional requirements concerning the file system operations are nowhere taught or suggested in the cited Harper, which only mentions adding elements to a linked list, not performing file system related operations as claimed.

Added claims 37, 38, and 39 depend from claims 1, 10, and 19 and further require that the function comprises a wide hash function to produce a large number of possible hash values to minimize the likelihood that the application of the hash function to file names in the file system would have a same hash value. The additional requirements of these added claims are disclose don pages 6-7 of the Application.

Applicants submit that added claims 37-39 are patentable over the cited art because they depend from one of claims 1, 10, and 19, which are patentable over the cited art for the reasons described above, and because the Examiner has not cited any art teaching the additional requirements of these added claims.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1-39 are patentable over the art of record. Applicants submit herewith the fees for the added claims and for a Petition for a One Month Extension of Time. Nonetheless, should any additional fees be required, please charge Deposit Account No. 50-0585.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: March 7, 2003

By: _____

David W. Victor
Reg. No.: 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes Victor & Mann, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984